

# CENG 3420

# Computer Organization & Design



## Lecture 14: Memory Organization-2

Bei Yu

CSE Department, CUHK

[byu@cse.cuhk.edu.hk](mailto:byu@cse.cuhk.edu.hk)

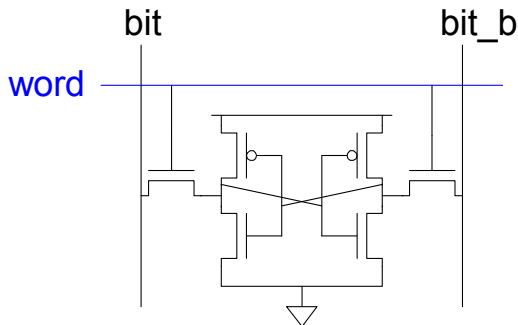
(Textbook: Chapters 5.1–5.2 & A.8–A.9)

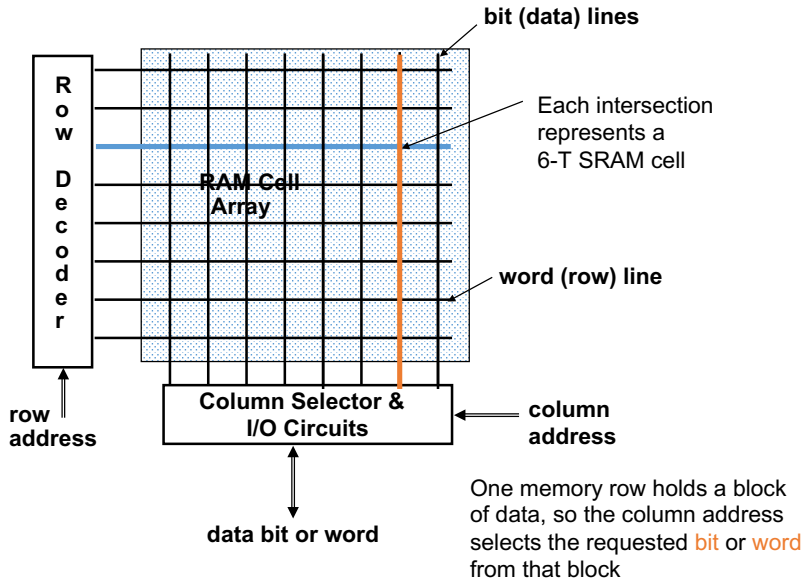
Spring 2022



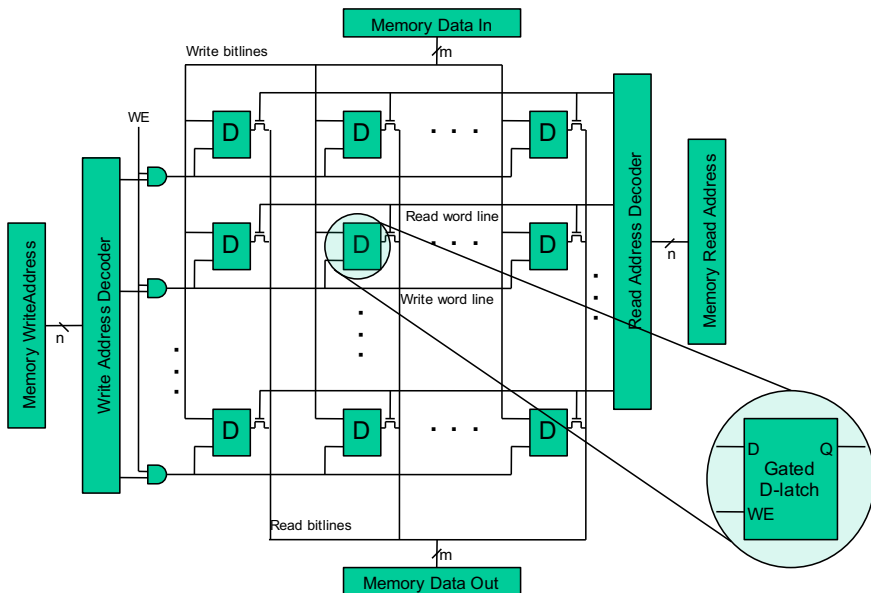
# Random Access Memory (RAM)

- At least 6 transistors (6T)
- Used in most commercial chips
- A pair of weak cross-coupled inverters
- Data stored in cross-coupled inverters

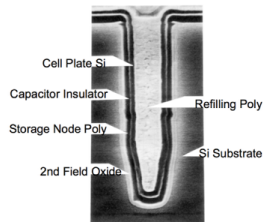
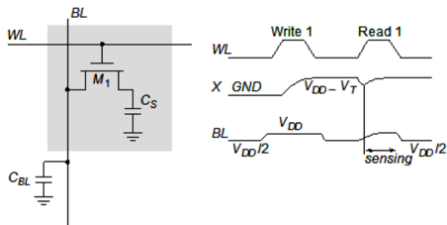




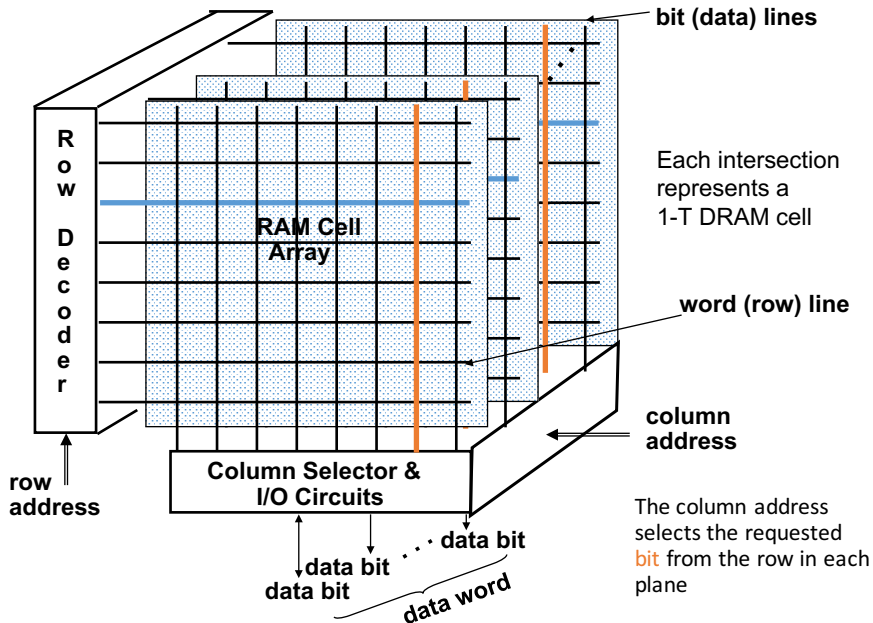
## Latch based memory



- 1 Transistor (1T)
- Requires presence of an extra capacitor
- Modifications in the manufacturing process.
- Higher density
- **Write:** Charged or discharged the capacitor (slow)
- **Read:** Charge redistribution takes place between bit line and storage capacitance



# Classical DRAM Organization – 3D Structure



- The common type used today as it uses a clock to synchronize the operation.
- The refresh operation becomes transparent to the users.
- All control signals needed are generated inside the chip.
- The initial commercial SDRAM in the 1990s were designed for clock speed of up to 133MHz.
- Today's SDRAM chips operate with clock speeds exceeding 1 GHz.

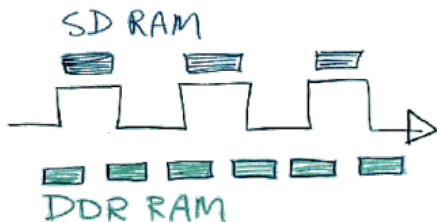
Memory modules are used to hold several SDRAM chips and are the standard type used in a computer's motherboard, of size like 4GB or more.







- normal SDRAMs only operate once per clock cycle
- Double Data Rate (DDR) SDRAM transfers data on both clock edges
- **DDR-2** (4x basic memory clock) and **DDR-3** (8x basic memory clock) are in the market.
- They offer increased storage capacity, lower power and faster clock speeds.
- For example, DDR2 can operate at clock frequencies of 400 and 800 MHz. Therefore, they can transfer data at effective clock speed of 800 and 1600 MHz.



1 Hertz

1 Cycle per second

RAM Type	Theoretical Maximum Bandwidth
SDRAM 100 MHz (PC100)	$100 \text{ MHz} \times 64 \text{ bit/cycle} = 800 \text{ MByte/sec}$
SDRAM 133 MHz (PC133)	$133 \text{ MHz} \times 64 \text{ bit/cycle} = 1064 \text{ MByte/sec}$
DDR SDRAM 200 MHz (PC1600)	$2 \times 100 \text{ MHz} \times 64 \text{ bit/cycle} \approx 1600 \text{ MByte/sec}$
DDR SDRAM 266 MHz (PC2100)	$2 \times 133 \text{ MHz} \times 64 \text{ bit/cycle} \approx 2100 \text{ MByte/sec}$
DDR SDRAM 333 MHz (PC2600)	$2 \times 166 \text{ MHz} \times 64 \text{ bit/cycle} \approx 2600 \text{ MByte/sec}$
DDR-2 SDRAM 667 MHz (PC2-5400)	$2 \times 2 \times 166 \text{ MHz} \times 64 \text{ bit/cycle} \approx 5400 \text{ MByte/sec}$
DDR-2 SDRAM 800 MHz (PC2-6400)	$2 \times 2 \times 200 \text{ MHz} \times 64 \text{ bit/cycle} \approx 6400 \text{ MByte/sec}$

Bandwidth comparison. However, due to latencies, SDRAM does not perform as good as the figures shown.



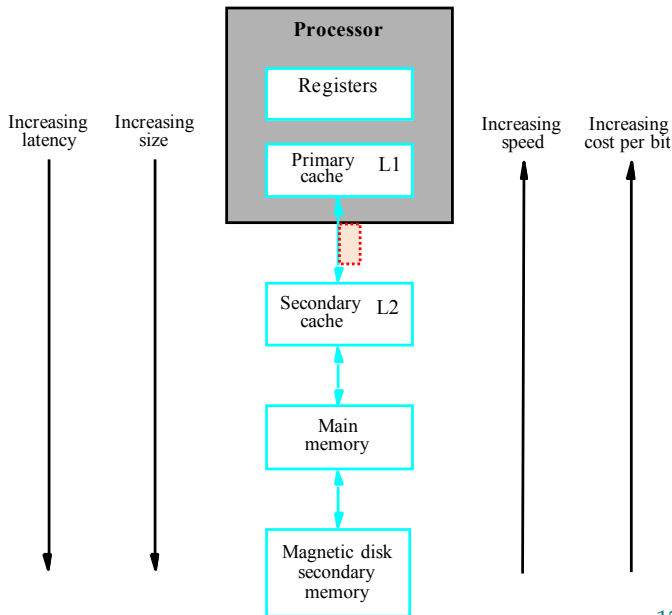
## Static RAM (SRAM)

- Capable of retaining the state as long as power is applied.
- They are **fast**, low power (current flows only when accessing the cells) but costly (require several transistors), so the capacity is small.
- They are the Level 1 cache and Level 2 cache inside a processor, of size 3 MB or more.

## Dynamic RAM (DRAM)

- store data as electric charge on a capacitor.
- Charge leaks away with time, so DRAMs must be refreshed.
- In return for this trouble, **much higher density** (simpler cells).

- **Aim:** to produce fast, big and cheap memory
- L1, L2 cache are usually SRAM
- Main memory is DRAM
- Relies on **locality of reference**





By taking advantages of the principle of locality:

- Present the user with as much memory as is available in the cheapest technology.
- Provide access at the speed offered by the fastest technology.

DRAM is **slow** but cheap and **dense**:

- Good choice for presenting the user with a BIG memory system – main memory

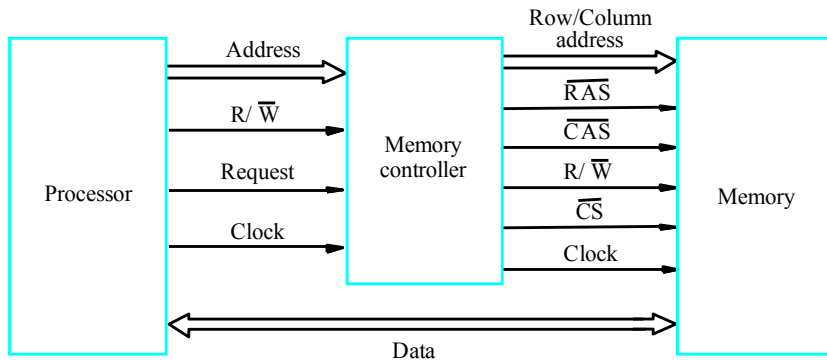
SRAM is **fast** but expensive and **not very dense**:

- Good choice for providing the user FAST access time – L1 and L2 cache



# Interleaving

- A **memory controller** is normally used to interface between the memory and the processor.
- DRAMs have a slightly more complex interface as they need refreshing and they usually have time-multiplex signals to reduce pin number.
- SRAM interfaces are simpler and may not need a memory controller.



RAS (CAS) = Row (Column) Address Strobe; CS = Chip Select



- The memory controller accepts a complete address and the R/W signal from the processor.
- The controller generates the **RAS** (Row Access Strobe) and **CAS** (Column Access Strobe) signals.

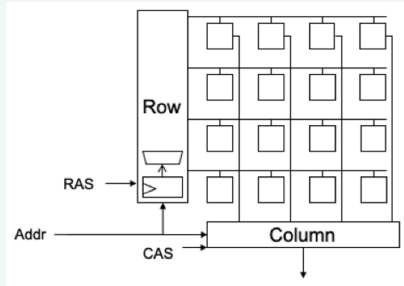




- The memory controller accepts a complete address and the R/W signal from the processor.
- The controller generates the **RAS** (Row Access Strobe) and **CAS** (Column Access Strobe) signals.
- The **high-order** address bits, which select a row in the cell array, are provided first under the control of the RAS (Row Access Strobe) signal.
- Then the **low-order** address bits, which select a column, are provided on the same address pins under the control of the CAS (Column Access Strobe) signal.



- The memory controller accepts a complete address and the R/W signal from the processor.
- The controller generates the **RAS** (Row Access Strobe) and **CAS** (Column Access Strobe) signals.
- The **high-order** address bits, which select a row in the cell array, are provided first under the control of the RAS (Row Access Strobe) signal.
- Then the **low-order** address bits, which select a column, are provided on the same address pins under the control of the CAS (Column Access Strobe) signal.
- The right memory module will be selected based on the address. Data lines are connected directly between the processor and the memory.
- SDRAM needs refresh, but the refresh overhead is only less than 1 percent of the total time available to access the memory.



- Put multiple words in one memory row – splits the decoder into two decoders (row and column)
- makes the memory core **square** reducing the length of the bit lines (but increasing the length of the word lines).
- This scheme is good only for up to 64 Kb to 256 Kb. For bigger memories it is too SLOW because the word and bit lines are too long.
- SRAM allows you to read an entire row out at a time at a word.

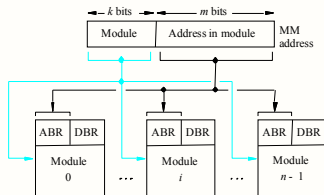


- Processor and cache are fast, main memory is slow.
- Try to hide access latency by **interleaving** memory accesses across several memory modules.
- Each memory module has own Address Buffer Register (**ABR**) and Data Buffer Register (**DBR**)

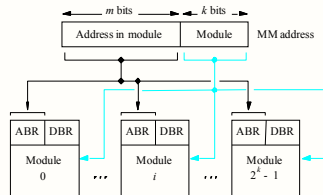


- Processor and cache are fast, main memory is slow.
- Try to hide access latency by **interleaving** memory accesses across several memory modules.
- Each memory module has own Address Buffer Register (ABR) and Data Buffer Register (DBR)

Which scheme below can be better interleaved?



(a) Consecutive words in a module

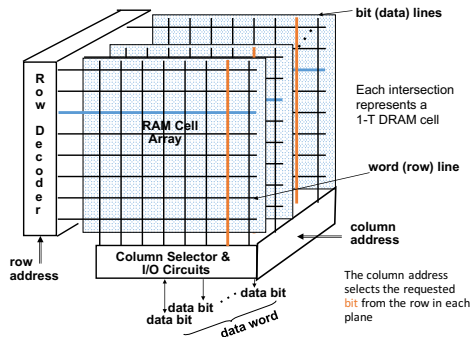


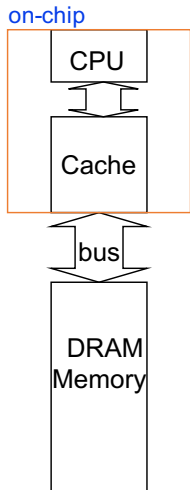
(b) Consecutive words in different modules



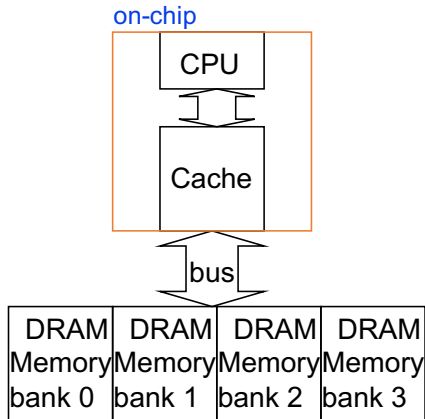
- (b) is better
- can provide parallel access.

- Two or more **compatible** (identical the best) memory modules are used.
- Within a memory module, several chips are used in “parallel”.
- E.g. 8 modules, and within each module 8 chips are used in “parallel”. Achieve a  $8 \times 8 = 64$ -bit memory bus.
- Memory interleaving can be realized in technology such as “**Dual Channel Memory Architecture**”.





(a)  
Non-Interleaving



(b) Interleaving

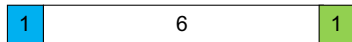


- Suppose we have a cache read miss and need to load from main memory
- Assume cache with 8-word block, i.e., cache line size = 8 words (bytes)
- Assume it takes **one clock** to send address to DRAM memory and **one clock** to send data back.
- In addition, DRAM has **6** cycle latency for first word
- Good that each of subsequent words in same row takes only **4** cycles

Single Memory Read:  $1 + 6 + 1 = 8$  Cycles

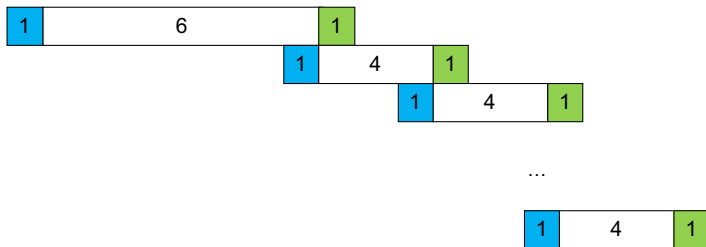


## Example: Non-Interleaving



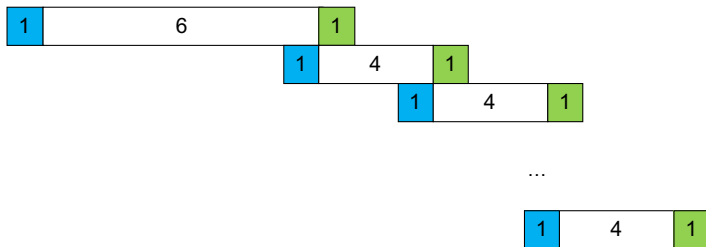
- First byte DRAM needs 6 cycle (same as single memory read)

# Example: Non-Interleaving



- First byte DRAM needs 6 cycle (same as single memory read)
- All subsequent words DRAM needs 4 cycle
- Non-overlappings in cache access
- **Assumption:** all words are in the same row

# Example: Non-Interleaving

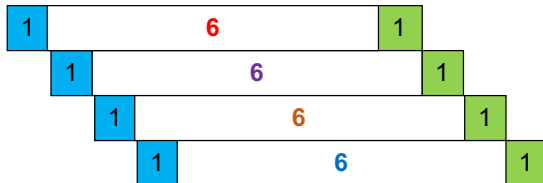


- First byte DRAM needs 6 cycle (same as single memory read)
- All subsequent words DRAM needs 4 cycle
- Non-overlappings in cache access
- **Assumption:** all words are in the same row

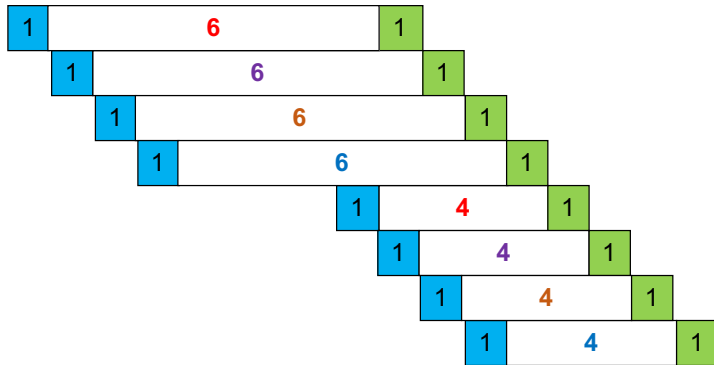
Non-Interleaving Cycle#

$$1 + 1 \times 6 + 7 \times 4 + 1 = 36$$

# Example: Four Module Interleaving



# Example: Four Module Interleaving



Interleaving Cycle#

$$1 + 6 + 1 \times 8 = 15$$



## Question:

To transfer 8 bytes, what is the cycle# if just have **TWO**-module interleaved?



## Formula for 2-module interleaving:

$$2 + 6 + 4 \times \left(\frac{n}{2} - 1\right) + 1 = 21$$

where  $n$  is the byte# to transfer.

- tip: you can draw such diagram using excel

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Blue							Green													
	Blue							Green												
						Blue					Green									
							Blue					Green								
										Blue					Green					
											Blue					Green				
														Blue					Green	
															Blue					Green





# Conclusion



- Processor usually runs much faster than main memory
- Common RAM types:  
SRAM, DRAM, SDRAM, DDR SDRAM
- Principle of locality: Temporal and Spatial
  - Present the user with as much memory as is available in the **cheapest** technology.
  - Provide access at the speed offered by the **fastest** technology.
- Memory hierarchy:
  - Register → Cache → Main Memory → Disk → Tape